

## Выполнение периодических заданий по крон

*Кульгун Екатерина Ивановна, Кульгун Александр Петрович*

УО «Брестский государственный университет имени А.С. Пушкина»

Допустим, нам нужно, чтобы определенная задача (функция нашего проекта) выполнялась на сервере периодически и фоново. Например, надо чтобы ночью собиралась статистика и отправлялась на веб-сайт, или чтобы клиентам рассылались письма с изменениями в их заказах, или чтобы каждые несколько часов обновлялись определенные данные в базе данных.

Периодические задачи - это функция, которая регулярно активирует скрипты и позволяет посещать страницы с определенной периодичностью. Это дает возможность выполнять разного рода задачи с заданным интервалом. При этом все происходит автоматически, и разработчику не приходится ничего делать вручную.

Утилита `cron` позволяет выполнять задание на регулярной основе с заданной периодичностью. Регулярные действия описываются инструкциями, помещенными в `crontab`-файлы каталога `/usr/spool/cron/crontabs`. В различных версиях GNU/Linux используются разные пакеты `cron`, работа которых отличается в деталях. Основой регулярного выполнения заданий `cron` являются таблицы, в которых кодируется периодичность выполнения заданий.

Система `cron` предназначена для запуска заданий по расписанию. Написан `cron` программистом Паулем Вики (Paul Vixie). Как и многие другие популярные программы, `cron` имеет множество инкарнаций. Больше того, существует множество разных `vixie-cron`'ов. В основном авторы клонов `cron`'а занимались тем, что добавляли те или иные удобства в синтаксис файла `crontab`.

`Cron` запускается каждую минуту и проверяет списки запланированных заданий - `crontab`, на предмет наличия там заданий, подлежащих запуску. `Crontab` это обычный файл, содержащий список команд, а также время, в которое эти команды должны запуститься.

`Cron` проверяет файлы с описанием регулярных действий и файлы команды `at` только во время инициализации процессов, а также при изменениях упомянутых файлов командами `crontab` и `at`. Это позволяет уменьшить накладные расходы на периодические проверки новых или измененных файлов.

Основные переменные `Cron`:

**SHELL** – эта переменная ответственна за то, в какой оболочке будут выполняться команды данного файла `crontab`. Теоретически сюда можно

подставить что-нибудь более экзотическое, например интерпретаторы perl или python.

**MAILTO** – в этой переменной должен упоминаться пользователь (или email) которому в виде почты будет отправляться стандартный вывод команд crontab'a. Если переменная определена, но пустая (MAILTO=""), то письмо не будет отправлено никому. Если же она просто не определена, письмо будет отправлено хозяину файла crontab или root'у если это системный crontab.

**PATH** – стоит заметить, что изначально эта переменная не определена. Поэтому команды лучше указывать с полными путями. Это хорошая практика так же и с точки зрения безопасности. Но имейте ввиду, что если вы вызываете shell-сценарий, то и в нём все команды должны отдаваться с абсолютными путями.

Далее следуют пять полей отвечающих за время, когда будет выполнено задание: минуты, часы, дни (месяца), месяца и дни недели. Касательно дней недели следует заметить, что их не 7, а 8: 0 и 7 соответствуют воскресенью. Таким образом, те кто считают, что первый день недели это понедельник, могут отсчитывать дни с единицы, а кто думает, что первый день недели это воскресенье — с нуля.

Особенности cron:

1. Все условия (времени запуска) проверяются по «логическому И», кроме условий «день недели» и «день месяца» — указанные совместно, они обрабатываются по «логическому ИЛИ», то есть «по любому из дней. Однако такая логика неочевидна и не позволяет создать условие типа «первый понедельник каждого месяца» или «каждую пятницу в 13 число». Разработчики не изменяют данное поведение, считая его стандартом.

2. Все файлы crontab должны завершаться пустой строкой, иначе будет проигнорировано последнее правило либо весь файл.

3. Команды файла crontab используют стандартный синтаксис командной оболочки, но их нельзя разделить на несколько строк.

4. Запускаемые файлы в автоматически сканируемых каталогах, таких как /etc/cron.daily/, игнорируются, если их имя содержит точку.

5. Синтаксис crontab, используемый для запуска раз в несколько минут со сдвигом, может привести к неожиданным результатам. Так, запись 1-56/5 \* \* \* \* не будет совершать запуск по 01-й минуте каждого часа; для срабатывания «каждые 5 минут начиная с 1-й минуты часа» следует использовать формат 1-59/5 \* \* \* \* или 1,6,11,16,21,26,31,36,41,46,51,56 \* \* \* \*.

6. Символ '%' в команде запуска используется для передачи перевода строки в команду (если не экранирован). Следующий за ним

текст команды передаётся запускаемой программе не как параметры командной строки, а через стандартный вход.

7. Вывод запускаемых команд отправляется пользователю, от имени которого производился запуск. Если в файле crontab до команды расположена директива MAILTO=<new@email>, то выводимые данные будут отправляться на указанный адрес.

## СПИСОК ЛИТЕРАТУРЫ

1. Успенский, В.А. Теория алгоритмов: основные открытия и приложения / В. А. Успенский, А. Л. Семенов. – М. : Наука, 1987. – 288 с.